

Práctica 2 – Base de datos, balanceo y escalado

Descripción de los pasos seguidos para cumplir los objetivos

Actividad 1:

Despliega dos instancia en EC2 con un servidor web que muestre una pagina similar pero que se pueda reconocer que es un servidor distinto. E.g. [El servidor de Gabriel 1] [El servidor de Gabriel 2]. Estos servidores deben poder ser accedidos con un navegador desde fuera.

1. Crear una primera instancia
2. Poner un nombre e.g.'Servidor_Javi_1'
3. Seleccionar SSH, HTTP & HTTPS para que sea un servidor web.

4. Insertar nuestro script en 'User data' para correrlo al inicio:

```
#!/bin/bash
yum update -y
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Servidor de Javi 1 </h1></html>' >
/var/www/html/index.html
```

User data - optional [Info](#)

Upload a file with your user data or enter it in the field.

 Choose file

```
#!/bin/bash
yum update -y
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Servidor de Javi 1 </h1></html>' >
/var/www/html/index.html
```

☐ User data has already been base64 encoded

5. Lanzar la instancia

6. Crear una segunda instancia

7. Poner un nombre e.g.'Servidor_Javi_2'

8. Seleccionar SSH, HTTP & HTTPS para que sea un servidor web.

Name and tags [Info](#)


Name


Servidor_Javi_2 [Add additional tags](#)


▼ Application and OS Images (Amazon Machine Image) [Info](#)


An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or browse for AMIs if you don't see what you are looking for below.


Quick Start


 Amazon Linux


 macOS

 Ubuntu

 Windows

 Red Hat

 SUSE Linux

 Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible

ami-03a6eaae9938c858c (64-bit (x86)) / ami-03f6c2c562b5d7715 (64-bit (ARM))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.2.20230920.1 x86_64 HVM kernel-6.1

Architecture

AMI ID

ami-03a6eaae9938c858c Verified provider

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select [Create new key pair](#)

▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-058cda2ce5f98b8ab

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

We'll create a new security group called 'launch-wizard-5' with the following rules:

☐ Allow SSH traffic from

Helps you connect to your instance

☒ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

9. Insertar nuestro script en 'User data' para correrlo al inicio:

```
#!/bin/bash
yum update -y
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Servidor de Javi 2 </h1></html>' >
/var/www/html/index.html
```

User data - optional [Info](#)
Upload a file with your user data or enter it in the field.

```
#!/bin/bash
yum update -y
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Servidor de Javi 2 </h1></html>' >
/var/www/html/index.html
```

☐ User data has already been base64 encoded

10. Lanzar la instancia

Actividad 2:

Despliega un “load balancer” que distribuya las peticiones entre los dos servidores a partes iguales.

Primero debemos tener creado un **Target Group**:

1. Seleccionar ‘Instancia’ como tipo de objetivo

Basic configuration
Settings in this section can't be changed after the target group is created.

Choose a target type

☒ **Instances**

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

2. Ponerle un nombre
3. Seleccionamos protocolo ‘HTTP’ y puerto ‘8080’
4. Seleccionar IPv4
5. Seleccionar ‘HTTP1’ como versión del protocolo

Target group name

PrimerTargetGroup

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port

HTTP : 8080

1-65535

IP address type

Only targets with the indicated IP address type can be included in this target group.

☒ IPv4

☐ IPv6

Each target you register must have an assigned primary IPv6 address. This is configured on the instances default network interface (eth0). [Learn more](#)

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

vpc-058cda2ce5f98b8ab
IPv4: 172.31.0.0/16

Protocol version

☒ HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

6. Elegimos las instancias que queremos registrar como objetivos (Nuestros Servidores)

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/2)

Filter resources by property or value

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups
<input checked="" type="checkbox"/>	i-00c7297b2724d9b2c	Servidor_Javi_1	Running	launch-wizard-5
<input checked="" type="checkbox"/>	i-05b5296549f90b12	Servidor_Javi_2	Running	launch-wizard-6

2 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

8080

1-65535 (separate multiple ports with comma)

Include as pending below

7. Crear el Target Group

Una vez creado, creamos nuestro **Load Balancer**:

1. Le ponemos un nombre e.g. **BalacerServidores1Y2'**
2. Seleccionamos **'Internet-facing'** como esquema.
3. Tipo de IP = **IPv4**

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

Balancer Servidores

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)
Scheme can't be changed after the load balancer is created.

☒ **Internet-facing**
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ **Internal**
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)
Select the type of IP addresses that your subnets use.

☒ **IPv4**
Recommended for internal load balancers.

☐ **Dualstack**
Includes IPv4 and IPv6 addresses.

4. Seleccionamos las áreas de disponibilidad (las que se encuentren las instancias)

Mappings [Info](#)
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☐ **us-east-1a (use1-az2)**

☐ **us-east-1b (use1-az4)**

☒ **us-east-1c (use1-az6)**

Subnet
subnet-046b25d65808e65e7

IPv4 address
Assigned by AWS

5. Configuramos Listeners -> **HTTP:8080** y nuestro **Security Group**

Listeners and routing [Info](#)
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener **HTTP:8080** [Remove](#)

Protocol **Port** **Default action** [Info](#)

HTTP : 8080
1-65535

Forward to **PrimerTargetGroup**
Target type: Instance, IPv4

HTTP [Create target group](#)

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)
You can add up to 50 more tags.

[Add listener](#)

6. En 'Health Check' seleccionar **HTTP** y 'actuador/health' com Path

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol
HTTP

Health check path
Use the default path of "/" to ping the root, or specify a custom path if preferred.
/actuator/health
Up to 1024 characters allowed.

► **Advanced health check settings**

Actividad 3:

Prepara un “template” de instancia para EC2 para generar servidores web. Con el “template” declarar un “Auto-Scaling Group”(ASG) que tenga como mínimo una instancia y como máximo 2. El ASG debe añadirse al “load balancer” previamente desplegado. Comprueba que el ASG mantiene al menos una instancia viva y que el “load balancer” le manda peticiones entrantes.

1. Creamos un nombre ‘MyTemplateWeb’ y elegimos su Verisón/Descripción
2. Seleccionamos ‘Auto Scaling Guidance’ para que nos ayude.

Launch template name and description

Launch template name - *required*
MyTemplateWeb
Must be unique to this account. Max 128 chars. No spaces or special characters like %, ", ' or @.

Template version description
Template de Servidor
Max 255 chars

Auto Scaling guidance [Info](#)
Select this if you intend to use this template with EC2 Auto Scaling
☒ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► Template tags
► Source template

Launch template contents
Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ **Application and OS Images (Amazon Machine Image) - required** [Info](#)

3. Seleccionar nuestro AMI -> ‘Amazon Linux’
4. Elegimos nuestro tipo de instancia -> ‘t2.micro’
5. Seleccionamos nuestro Key Pair -> ‘labusers.pem’

Quick Start

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible

ami-03a6eaae9938c858c (64-bit (x86)) / ami-03f6c2c562b3df715 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.2.20230920.1 x86_64 HVM kernel-6.1

Architecture 64-bit (x86) **AMI ID** ami-03a6eaae9938c858c Verified provider

▼ Instance type Info Advanced

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand RHEL base pricing: 0.0716 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

[Additional costs apply for AMIs with pre-installed software](#)

☐ All generations [Compare instance types](#)

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.


Key pair name

labusers Create new key pair

6. Añadimos nuestro 'User Data'

```
#!/bin/bash
yum update -y
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Servidor de Javi <Numero </h1></html>' >
/var/www/html/index.html
```

User data - *optional* [Info](#)
 Upload a file with your user data or enter it in the field.

 Choose file

```
#!/bin/bash
yum update -y
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Servidor de Javi <Numero> </h1></html>' >
/var/www/html/index.html
```

☐ User data has already been base64 encoded

7. Lanzamos Template

Ahora debemos crear un “Auto-Scaling Group”(ASG):

1. Primero asignar un nombre e.g.'ASG_WebServer'
2. Seleccionar nuestro Template y su versión


Name

Auto Scaling group name
 Enter a name to identify the group.


Must be unique to this account in the current Region and no more than 255 characters.

Launch template [Info](#)

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
 

[Create a launch template](#)

Version
 

[Create a launch template version](#)

Description Template de Servidor	Launch template MyTemplateWeb lt-09f6dc1aa577392f9	Instance type t2.micro
AMI ID ami-03a6eaae9938c858c	Security groups -	Request Spot Instances No
Key pair name labusers	Security group IDs sg-0ef034c473c3ce97a	

Additional details

Storage (volumes) -	Date created Sat Sep 30 2023 11:57:12 GMT+0100 (Western European Summer Time)
------------------------	---

3. Seleccionamos VPC 'default' y nuestras zonas de disponibilidad

Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-058cd2ce5f98b8ab
172.31.0.0/16 Default

[Create a VPC](#)

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-1c | subnet-046b25d65808e5e7 ✕
172.31.32.0/20 Default

us-east-1d | subnet-098581d79edb517d1 ✕
172.31.0.0/20 Default

[Create a subnet](#)

Instance type requirements [Info](#)

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

[Override launch template](#)

Launch template	Version	Description
MyTemplateWeb lt-09f6dc1aa577392f9	Default	Template de Servidor

Instance type
t2.micro

4. Para el Load Balancer seleccionamos ‘Existente’, ‘Elegir de Target Groups’ y elegir nuestro **Target Group**.

Load balancing [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

☒ Attach to an existing load balancer
Choose from your existing load balancers.

☐ Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.

☒ Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

☐ Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

PrimerTargetGroup | HTTP ✕
Application Load Balancer: BalancerServidoresTY2

VPC Lattice integration options [Info](#)

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

☒ No VPC Lattice service
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

☐ Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

[Create new VPC Lattice service](#)

5. Elegir ‘Elastic Load Balancer health checks’

Health checks
Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks
[Always enabled](#)

Additional health check types - *optional* [Info](#)

☒ **Turn on Elastic Load Balancing health checks** Recommended
Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

[Info](#) EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the [Load Balancer console](#) [console](#) [X](#)

☐ **Turn on VPC Lattice health checks**
VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

Health check grace period [Info](#)
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

seconds

6. Para el tamaño de grupo seleccionamos:

- a. 1 como *Óptimo*
- b. 1 como *Mínimo*
- c. 2 como *Máximo*

Group size - optional [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity

Minimum capacity

Maximum capacity

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

☐ **Target tracking scaling policy**
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

☒ **None**

Instance scale-in protection - optional

Instance scale-in protection
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

☐ **Enable instance scale-in protection**

7. Creamos nuestro **ASG**

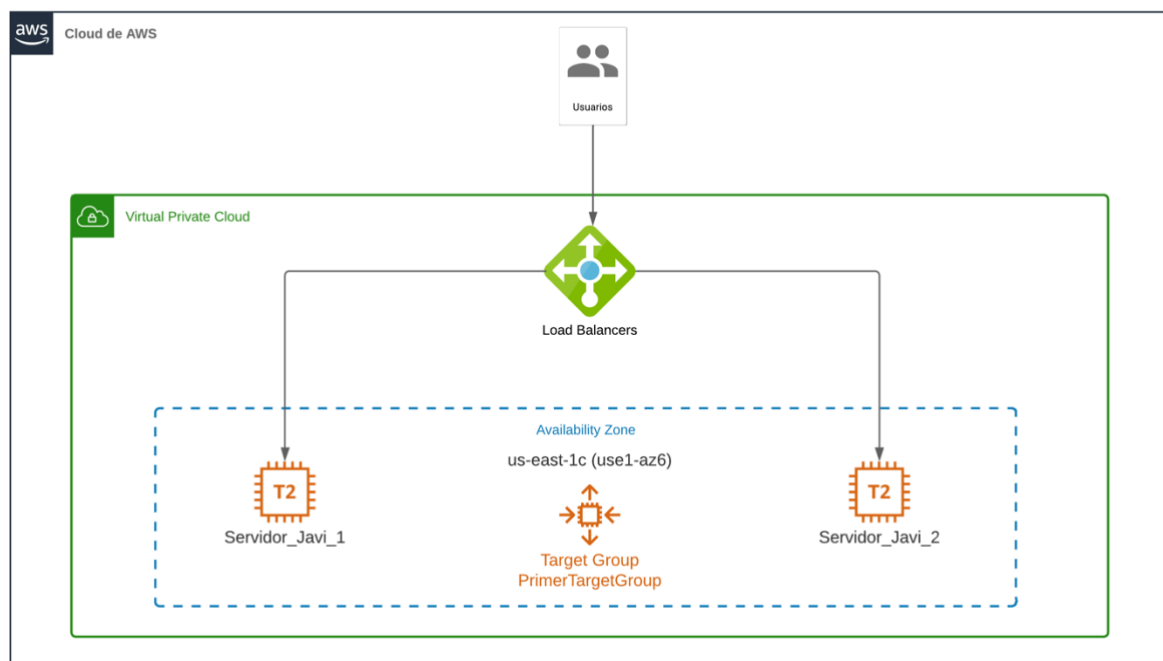
Actividad 4:

Diagrama de arquitectura desplegada

Herramienta usada: [Lucidchart](https://lucidchart.com)






El diagrama muestra dos instancias de tipo 't2.micro'.
Las instancias son servidores webs llamados 'Servidor_Javi_1' y 'Servidor_Javi_2'.

Las instancias tienen los puertos 80 y 443 para tener el HTTP y HTTPS activos para actuar como servidores web. En el diagrama se muestran los 'Load Balancers', 'Target Groups' y 'Zonas Disponibles'.



Presupuesto y estimación de gasto de los recursos desplegados

Durante la práctica, el coste generado ha sido \$0.28 USD:

Cost and usage Info	
Current month costs	Top costs for current month
\$0.32	 EC2 - Other \$0.31
Forecasted month end costs	 Amazon Elastic Compute Cloud - C... \$0.01
-	 AmazonCloudWatch \$0.00
Last month costs	 CloudWatch Events \$0.00
\$0.00	 AWS CloudShell \$0.00
Costs shown are unblended. Learn more	

Según la estimación de gastos para nuestra instancia de tipo 't2.micro':

On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Para nuestro tipo de instancia, mensualmente(720 Horas) en un Windows nos costaría alrededor de unos \$11,664 USD. Este precio se da si mantenemos el servidor web en línea todos los días, sin contar tiempo de mantenimiento ni reparaciones.

El precio de mantener los 2 servidores sería \$23,328 USD. El balanceador de carga son \$0,0252 USD a la hora, lo cual mensualmente saldría unos \$70,308 USD.